```
module alu(A,B,SELECT,RESULT,E);

input [15:0] A,B;
input SELECT;
output [15:0] RESULT;
output E;

reg [15:0] RESULT ;
wire [16:0] sum;


assign sum = {1'b0,A} + {1'b0,B};
assign E = sum [16];

always @(*)
begin
case(SELECT)
1'b0:
RESULT = A & B;
1'b1:
RESULT = A + B;
default: RESULT = A + B ;
endcase
end

endmodule


module alu_tb (A,B,SELECT,RESULT,E);

output [15:0] A,B;
output SELECT;
input [15:0] RESULT;
input E;

reg[15:0] A,B;
reg SELECT;
wire[15:0] RESULT;
wire E;

initial
begin
$dumpfile ("alu_tb.vcd");
$dumpvars;
A = 16'b1111111111001100;
B = 16'b1111100011111111;
SELECT = 1'b0;
#100 $finish;
end
```

```
initial
begin
#70 SELECT=1'b1;
end

initial
begin
#50 A=16'b1100111110000011;
end

initial
begin
#50 B=16'b0011100010001001;
end

alu A1 (A,B,SELECT,RESULT,E);


endmodule
```